

REMARKS

This Amendment is in response to the Office Action dated February 11, 2005, in which claims 1-33 were initially rejected. Applicants respectfully request reconsideration and allowance of all pending claims in view of the above-amendments and the following remarks.

I. PRIORITY

The Office Action acknowledges Applicant's claim for foreign priority, but suggested that Applicant has not filed a certified copy of the French Application No. 0012494. However, Applicants submitted a certified copy on December 20, 2001.

Enclosed with this amendment is a copy of the self-addressed return postcard indicating that the certified copy was submitted on December 20, 2001 and received by the Patent Office on January 24, 2002. Also enclosed is a copy of the cover letter submitted with the priority document, and a photocopy of the front page of the priority documents.

However, Applicant's attorney is aware of significant PTO mail problems during that time period due to extraordinary mail screening measures.

If the Patent Office requires an additional certified copy of the priority document, Applicants respectfully request a further notice requesting such copy. In that event, a new copy will be obtained and submitted to the PTO.

II. DOUBLE PATENTING

Claims 1-33 were provisionally rejected under the judicially created doctrine obviousness-type double patenting as being unpatentable over claims 1-31 of copending Application Number 09/965,674.

Enclosed with this Amendment is a Terminal Disclaimer, which is believed to overcome the obviousness-type double patenting rejection.

III. CLAIM REJECTIONS UNDER §112

Claims 12 and 13 were rejected under §112, second paragraph as being indefinite due to a typographical error. Accordingly, claim 12 is amended to correct the typographical error.

In addition, the specification is amended to correct a typographical error on page 17, line 32. These amendments do not involve the addition of any new matter and are not made in view of any prior art reference.

IV. CLAIM REJECTIONS UNDER §103

Claims 1-22 and 28-33 were rejected under §103(a) as being unpatentable over Angles et al., U.S. Patent No. 6,385,592 in view of Landsman et al., U.S. Patent No. 6,314,451.

A. Independent Claim 1

Claim 1 includes, "an on-the-fly insertion step of at least one active code in said page by at least one component of the architecture of said network," and "the area of said page wherein said active code is inserted is determined according to the type of action generated by said active code" Claim 1 further includes the limitation, "said active code inserted on the fly is an intermediate invocation active code, which, when run by said terminal, enables said terminal, during an invocation step, to invoke a final active code provider, so that the terminal receives from said provider a specific final active code enabling the running of an algorithm on said terminal."

1. Angles et al.

On page 5, the Office Action correctly acknowledges that Angles does not disclose these elements. In fact, Angles supposes that the content provider has embedded in the returned web page an embedded advertisement request (see Angles, FIG. 3). Therefore, there is no intermediate network equipment, providing any added value in Angles. In practice, when the returned web page is loaded by the browser located on the end user computer,

the returned web page sends a request to the web server belonging to the advertisement company.

2. Landsman et al.

Landsman, on the contrary to what is asserted by the Office Action, also does not disclose and on-the-fly insertion step. This can be seen from example in the Abstract where Landsman states, "Specifically, an HTML advertisement tag is embedded into a referring web page."

Fig. 2A shows the content of the referring contents page, which is the page that is computed, built and returned by the content provider. As the referring contents page is run by the end-user browser, the page gets contents that are formed into another format containing some applet code. Landsman is quite clear at explaining that this close formation occurs on the end-user browser as a result of the downloaded piece of code, which resulted from the execution of the initial referring contents page having an embedded tag.

As a result, there is no active code inserted by an intermediate network component in either Angles or Landsman. Thus, even if the disclosures of Angles and Landsman were combined as suggested in the Office Action, the resulting combination would still fail to teach or suggest each and every element of independent claim 1.

B. Dependent Claims 2-22 and 28-30

Many of the remaining claims are focused on the presence and/or use of this intermediate equipment, such as how a code can be inserted in a page, which is not prepared for by the content provider. For example, some claims are directed to what type of code can be inserted and how this can be combined with the use of HTTP cookies to retrieve and transmit information about the end-user to the advertisement company without a need for a registration mechanism.

Referring to the numbered paragraphs in the Office Action, many of the Examiner's points are separately addressed below:

Point 10 (Claim 2)

Although Angles discloses a communication network can include the Internet, Applicant believes the invention recited in claim 1 is fairly different. So it is reasonable to protect its applicability to an Internet network as per the claim 2, which in addition is dependent on claim 1.

Point 11 (Claim 3)

Similarly although Angles discloses that at least one of the components of the architecture of said network belongs to an element of the group comprising: Internet site host servers, access provider equipment, service provider equipment, routers, switches, getaways and proxies, Applicant believe the invention recited in claim 1 is fairly different. So it is reasonable to protect its applicability to such equipment as it is recited in claim 3.

Point 12 (Claim 4)

Angles as we have shown before, does not rely on an active code inserted by an intermediate network component. It supposes that the content provider has specifically prepared the returned web page so that its execution on the end-user browser will cause a request to be sent to the advertisement server. Column 16, lines 13-24, describes the function of the "advertisement provider computer 18". This is the computer that executes the request sent by the end-user browser generated as a result of the downloaded page. Typically, it is a CGI script and this computer acts as a regular web server. In claim 4 on the other hand, we recited that the active code is running on said terminal and thus is definitely not running on the advertisement server.

Point 13 (Claim 5)

Angles describes the use of a cookie, in conjunction with its application. It explains that the cookie is loaded by the end-user on its terminal during a registration phase. This is triggered by the step 708 described in FIG. 7. Column 19, line 55, shows that the cookie aims at returning the consumer member code whenever an advertisement request with the active code is sent by the end-user browser to the advertisement company web server. This is a traditional use of a cookie.

Claim 5 is more complex, and this is illustrated in FIG. 6, for example, of the present application. In this embodiment, the proposed mechanism will change the final active code based on the content of the cookie that was initially delivered. In addition to explaining the potential role played by cookies with the scope of the application, which is different from Angle's one, the role that described in claim 5 goes beyond the usage done by Angles in his application.

Point 14 (Claim 6)

The Office Action that "Angles discloses at least one step, prior to said on-the-fly insertion step and belonging to the group comprising: (a) steps consisting of rerouting an access of said terminal to the final active code provider, (b) steps consisting of defining a profile of the user of said terminal, (c) steps consisting of generating said at least one cookie as a function of the user profile of said terminal, (d) steps consisting of provision of said at least one cookie as a function of the user profile of said terminal, (d) steps consisting of provision of said at least one cookie by the final active code provider to said terminal, and (e) steps consisting of storage of said at least one cookie by said terminal."

Regarding element (a), we already saw that Angles does not rely on an intermediate communication network component for inserting an active code. Concerning b, c, d and e, we also already saw that Angles' application is quite different from that

recited in claim 1. Both systems may make use of the standard HTTP cookies, but according to different ways. So a claim stating how the new application can take advantage of this should be allowable. Angles in column 8, at lines 19-30, mentions that cookies are expected to carry a member code. This makes sense in the context of Angles' application. As the advertisement server handles both the registration and the CGI scripts user later on, the member code is enough to retrieve from a local database all the registration information. In the present application, we are not so restrictive since the script server can receive registration information directly as part of the cookie. In particular, it is not specified that the script server belongs to an advertisement company. As shown at FIG. 6, it could belong to the portal, it also could belong to an advertisement company independent of the portal, or it could belong to the same company as the Web server. Passing registration information via the cookie allows the active code to rely on the membership with a company but not necessarily with the advertisement company.

Point 16 (Claim 8)

Again, Angles does not rely on active code insertion in the web page independently of the content provider. In fact, for a good working of Angles it is likely that the content provider enters into a contractual relationship with the advertisement company. The method disclosed in the present application may work without such a requirement. Any content can thus be modified to receive advertisement or other types of information. A contract for the purpose may be agreed between the end-user and the Internet service provider. This type of relationship is not at all covered by Angles application, while this is however fully under the scope of the present application.

Point 17 (Claim 9)

In column 7, lines 7-18, Angles merely describes a list of terms an acronyms used throughout the description. Column 5,

line 61, provides a list describing the Web state-of-the-art. Angles does not insert on-the-fly active code into the content provider returned web pages.

Point 18 (Claim 10)

As in the previous case, Angles merely describes the Web state-of-the-art. At column 6, lines 44-49 in particular is described the role of the web browser, which indeed is to download and parse an HTML document. This document can include some hyper-links causing further network requests for additional content required to complement what needs to be displayed on the end user screen. The notion of active code goes beyond the function of the web browser. It can include a program, which does not parse the content according to the HTML grammar, that according to its own specific criterions generally in order to decide what type of content should be downloaded to add further information to the displayed page. In HTML, the decision is explicitly made by the content provider, while in the present application, it can explicitly be made by the company inserting the active code.

Point 19 (Claim 11)

In Angles, there is no active code involved. The content provider embeds standard HTML commands into the return page. These embedded commands initiate a new request to the advertisement web server, where a CGI script will perform some function. The advertisement web server CGI scripts do not run on the end-user terminal.

Point 20 (Claim 12)

The Angles application does not involve a step comprising generating an active code that will be inserted in the returned web page on the fly before it reaches the end-user terminal. In Angles, the page is prepared by the content provider to contain an HTML command that is interpreted by the browser in order to send a request to the advertisement web

serve, where a script will compute what would be displayed in the reserved space on the end-user terminal. These are obviously very different mechanisms to control the end-user display.

Point 21 (Claim 13)

Although the present application and Angles are very different in the way they operate, they have in common a high-level goal, which is to add the best to the content selected for the end-user, based on information obtained about him beforehand. However, Applicant should be allowed to protect the specific mechanism for achieving that goal, as recited in claim 13 within the context of claim 1.

Point 22 (Claim 14)

Claim 14 covers various ways to take advantage of the mechanism described in claim 1 to provide value-added services to the end-user. Since the method of claim 1 is very different from Angles, the applications recited in claim 14 are likewise patentable over Angles.

Point 23 (Claim 15)

In column 19, lines 64-67, Angles describes how the code running on the advertisement web server can take advantage of the cookie to deliver better content (such as focused advertisement) to the end-user.

Claim 15, specifies that the active code when it runs on the end-user terminal, can parse the page being read, to request specific information to a third-party server. The Examiner can note that the method recited in claim 15 is more powerful since Angles cannot decide what to return to the end-user page based on the member code information to decide that content to return. It doesn't know the content of the page from which the generated advertisement request came.

Point 24 (Claim 16)

Although the present application and Angles are very different in the way they operate, they have in common a high-

level goal which is to add information such as advertisement connotations or others to the displayed page, possibly dependent on user profile information obtained about him beforehand. However due to these differences, Angles does not disclose the elements of claim 16 within the context of claim 1.

Point 25 (Claim 17)

Claim 17 requires presentation of at least part of said downloaded data according to a format defined by the information provider in response to said specific information list provision operation.

Landsman supposes that the content provider has embedded some specific HTML request into its page. The inserted code causes the dot allowed by the end user browser of a Java applet, which modifies the displayed web page code this installs a Java agent on the end-user terminal, which then starts to download advertisements as a background task. It is then capable to display advertisement between two end-user web requests. The mechanism described by Landsman does not modify the presentation of the page, based on specific information searched in the returned page. In fact, its goal is to maintain advertisement completely independent of the referring web page. All that Landsman needs is the inserted tag to trigger the download of its Java agent. The Java agent generates a list of files which would be stored on the end-user terminal, for advertisement purposes. This is described in column 13, lines 55-67 and others. In short, Landsman builds a cache of advertisement pages on the end-user terminal, and Angles relies on a remote system to determine advertisement information to be added to a page.

It is not obvious to derive from these two systems the method recited in claim 17, where a new component is involved and where the decision of what must be added to the displayed web page can be made dynamically by looking at the content of the page.

Point 26 (Claim 18)

Applicants have not been able to find in Landsman evidence of searching for specific information in the page, creation of a list of the information found on the page, provision of the list of an information provider or censoring at least part of the downloaded data according to at least one criterion defined by the information provider in response to a specific information list provision operation.

Landsman modifies the page, such that a code can detect information between pages (clicks) and describes that the downloaded applet application preloads a list of advertisements to be inserted whenever a transition is detected. These are clearly two different mechanisms.

Point 27 (Claim 19)

Based on similar arguments discussed with respect to claim 18 and other claims, claim 19 is non-obvious over the prior art as well.

Point 28 (Claim 20)

As described at point 26, neither Landsman nor Angles, to that effect, present an active code that analyses the returned page, to create a list of specific information and to pass this list to a content provider. As a result, claim 20 is allowable over the cited references.

Point 29 (Claim 21)

Claim 21 is also closely tied to the roll of an active code parsing the page to create a list of specific information. As this is not disclosed by either Landsman or Angles, this claim should be sustained.

Point 30 (Claim 22)

Claim 22 gives some precision on what specific criterion can be used to update dynamically the specific information collected by the active code. As the notion of active code described in the present application is not disclosed

by either Landsman or Angles, this claim should be sustained as well.

Point 31 (Claim 28)

Neither Landsman nor Angles discloses an on-the-fly insertion step. Therefore the claim that this step can be systematic or selective should be sustained. We note the column 20 lines 30-38 in Angles, describes the fact that the end-user can click on the advertisement and that sends a request to the advertisement web server. It is clear that this is a step occurring well after the processing involved in this election of the advertisement that should be included in the returned page. It is thus not relevant to this particular topic.

Point 32 (Claim 29)

Again, neither Landsman nor Angles discloses an on-the-fly insertion step. Therefore the claim that the active code can be interrupted when the user requests the display of a new data page should be sustained. We note that column 2, lines 8-16 of Angles, describes the standard HTML hyperlink mechanism. It is not relevant to the notion of an active code execution model.

Point 33 (Claim 30)

As described above, Landsman does not disclose insertion of an active code within the context of claim 1. Thus, claim 30 is allowable with the allowance of claim 1.

In summary, dependent claims 2-22 and 28-30 are neither taught nor suggested by Angles or Landsman. Accordingly, Applicants respectfully request that the rejection of claims 1-22 and 28-30 under §103 be withdrawn.

C. Claims 31-33

Claims 31 and 32 are independent claims, which include similar limitations as were discussed above with respect to claim 1 that are neither taught nor suggested by Angles and Landsman. Claim 33 depends from claim 32.

Applicants therefore respectfully request that the rejection of claims 31-33 under §103(a) based on Angles and Landsman be withdrawn.

D. Claims 23-27

Claims 23-27 were rejected under §103(a) as being unpatentable over Angles et al. in view of Landsman et al. and further in view of Hawkins et al.

The Office Action correctly acknowledges that Angles and Landsman do not disclose at least one application for permanent provision to the user, via said page of at least one service provided by at least one fourth remote site connected to said network, and in that said active code, when it is run by the terminal, declares said at least one service in said page.

Hawkins describes a broadcasting mechanism allowing to build a programming guide out of information coming from various media objects. This is an environment totally different from the web and HTLM/HTTP protocols of Angles and Landsman, so the proposed combination of Hawkins would not be obvious to a person of ordinary skill in the art. Further, it is not clear how Hawkins relates to claims 23-27.

Hawkins describes a way to propose streaming events to end-users. Using a remote control, the end-user can select one of several options provided by an open "media valet" application. Note that Hawkins does not include any insertion of active code into one of the video streams.

In fact, Hawkins builds a new video stream by combining server sources. There is no notion of an active code modifying a return video stream (as sent by a content provider), parsing it to detect some information elements and modifying the stream itself to insert bits of information such as advertisement. Dependent claim 23 relies on a notion of active code. Since the video streams of Hawkins have nothing to do with an active code, the resulting combination of Angles, Landsman and Hawkins would

still fail to teach or suggest each and every element of claim 23. Further, it is unclear how such a modification could be made, based on the teachings of the cited references.

With respect to claim 24, Hawkins describes in column 11, lines 40-58 how a menu could look like. The point of claim 24 is that an access menu can be implemented by the terminal within the context of the invention recited in claims 1 and 23. Such an implementation of an access menu is neither taught nor suggested by Hawkins.

Similar deficiencies with Hawkins exist with respect to dependent claims 25-27, Applicants therefore respectfully request that the rejection of claims 23-27 based on Angles, Landsman and Hawkins be withdrawn.

The Director is authorized to charge any fee deficiency required by this paper or credit any overpayment to Deposit Account No. 23-1123.

Respectfully submitted,

WESTMAN, CHAMPLIN & KELLY, P.A.

By: D. D. Brush

David D. Brush, Reg. No. 34,557
Suite 1400 - International Centre
900 Second Avenue South
Minneapolis, Minnesota 55402-3319
Phone: (612) 334-3222 Fax: (612) 334-3312

DDB:tkj